

WHAT IS CLAIMED IS:

1. A system for distributed automated software graphical user interface (GUI) testing, comprising:

a centralized test queue operable to store a plurality of software GUI test instances to be executed by a plurality of distributed test execution computers, each distributed test execution computer comprising a client platform and coupled to one or more server platforms, the client platforms and server platforms collectively providing a plurality of client-server combinations against which the software GUI test instances may be executed; and

10 a test server engine operable to, for each distributed test execution computer:

receive a request for a software GUI test instance from a particular distributed test execution computer in response to completion of a preceding software GUI test instance by the particular distributed test execution computer;

15 retrieve a software GUI test instance from the test queue;

communicate the retrieved software GUI test instance to the particular distributed test execution computer for execution against a particular client-server combination using a testing component supported by the particular distributed test execution computer, the testing component operable to perform automated software GUI testing and to produce test results for such testing for communication to the test server engine;

20 receive a test result for the software GUI test instance from the particular distributed test execution computer in response to execution of the software GUI test instance; and

25 store the received test result for reporting to one or more users.

2. The system of Claim 1, wherein at least one distributed test execution computer operates at a location geographically remote from the other distributed test execution computers and from the test server.

30 3. The system of Claim 1, wherein the testing component is a commercial off-the-shelf product.

4. The system of Claim 1, wherein each software GUI test instance is an instance of a software GUI test written using a test scripting language and can be executed using any of the distributed test execution computers, a software GUI test instance being executed using the particular distributed test execution computer from which the request initiating retrieval of the software GUI test instance from the test queue was received.

5. The system of Claim 1, wherein:

the test server engine is further operable to generate a test results web page comprising test results for a plurality of software GUI test instances, including the test result for the most recently executed software GUI test instance, substantially immediately upon receiving the test result from the particular distributed test execution computer on which the most recently executed software GUI test instance was executed; and

10 the system further comprises a web server operable to communicate the test results web page for display on a user system to provide substantially real-time test results reporting.

15 6. The system of Claim 5, wherein:

each software GUI test instance is an instance of a software GUI test; and
20 the test results web page comprises consolidated test results for a particular client platform, the consolidated test results indicating test results for each software GUI test for each client-server combination involving the particular client platform.

25 7. The system of Claim 5, wherein the test server engine is further operable to receive a user request to execute an instance of a particular software GUI test and to insert the requested software GUI test instance into the test queue according to the user request, the user request being input by selecting the particular software GUI test using the test results web page.

30 8. The system of Claim 1, wherein at least some GUI test instances in the test queue have associated priorities, the test server engine operable to retrieve the

GUI test instances from the test queue for execution according to their associated priorities.

9. The system of Claim 1, wherein the test queue comprises a first queue containing higher priority software GUI test instances and a second queue containing lower priority software GUI test instances, the test server engine operable to retrieve higher priority software GUI test instances from the first queue for execution during a first part of a testing period and retrieve lower priority software GUI test instances from the second queue for execution during a second part of the testing period.

10 10. The system of Claim 1, wherein the test server engine is operable to re-communicate instances of a software GUI test for execution against all client-server combinations, according to a rule, in response to receiving one or more test results for the software GUI test indicating failure.

15 11. The system of Claim 1, wherein the test server engine is operable to detect when the number of software GUI test instances in the test queue is below a predefined threshold and, in response, to automatically add software GUI test instances to the test queue.

20 12. The system of Claim 1, further comprising a client controller associated with each distributed test execution computer and operable to automatically install a current software GUI build at each distributed test execution computer at one or more appropriate times during a testing period.

25 13. The system of Claim 1, further comprising a client controller associated with each distributed test execution computer and operable to automatically reboot each distributed test execution computer according to a predetermined schedule.

30 14. The system of Claim 1, further comprising a client controller associated with each distributed test execution computer and operable to establish

communication with the test server engine when the distributed test execution computer boots up.

15. The system of Claim 1, wherein each test execution computer operates
5 essentially as an automated test execution robot, repeatedly requesting, receiving,
. executing, and returning test results for software GUI test instances, automatically and
without human intervention, for an extended time period.

16. The system of Claim 1, further comprising the distributed test
10 execution computers.

17. A method for distributed automated software graphical user interface (GUI) testing, the method comprising:

5 maintaining a centralized test queue operable to store a plurality of software GUI test instances to be executed by a plurality of distributed test execution computers, each distributed test execution computer comprising a client platform and coupled to one or more server platforms, the client platforms and server platforms collectively providing a plurality of client-server combinations against which the software GUI test instances may be executed; and

10 receiving, for each distributed test execution computer, a request for a software GUI test instance from a particular distributed test execution computer in response to completion of a preceding software GUI test instance by the particular distributed test execution computer;

15 retrieving, for each distributed test execution computer, a software GUI test instance from the test queue;

20 communicating, for each distributed test execution computer, the retrieved software GUI test instance to the particular distributed test execution computer for execution against a particular client-server combination using a testing component supported by the particular distributed test execution computer, the testing component operable to perform automated software GUI testing and to produce test results for such testing;

25 receiving, for each distributed test execution computer, a test result for the software GUI test instance from the particular distributed test execution computer in response to execution of the software GUI test instance; and

storing, for each distributed test execution computer, the received test result for reporting to one or more users.

30 18. The method of Claim 17, wherein at least one distributed test execution computer operates at a location geographically remote from the other distributed test execution computers and from a computer system on which the method is performed.

19. The method of Claim 17, wherein the testing component is a commercial off-the-shelf product.

20. The method of Claim 17, wherein each software GUI test instance is an instance of a software GUI test written using a test scripting language and can be executed using any of the distributed test execution computers, a software GUI test instance being executed using the particular distributed test execution computer from which the request initiating retrieval of the software GUI test instance from the test queue was received.

21. The method of Claim 17, further comprising:

10 generating a tests results web page comprising test results for a plurality of software GUI test instances, including the test result for the most recently executed software GUI test instance, substantially immediately upon receiving the test result from the particular distributed test execution computer on which the most recently executed software GUI test instance was executed; and

15 communicating the test results web page for display on a user system to provide substantially real-time test results reporting.

22. The method of Claim 21, further comprising:

each software GUI test instance is an instance of a software GUI test; and

20 generating a test results web page comprising consolidated test results for a particular client platform, the consolidated test results indicating test results for each software GUI test for each client-server combination involving the particular client platform.

25 23. The method of Claim 21, further comprising receiving a user request to execute an instance of a particular software GUI test and inserting the requested software GUI test instance into the test queue according to the user request, the user request being input by selecting the particular software GUI test using the test results web page.

30 24. The method of Claim 17, wherein at least some software GUI test instances in the test queue have associated priorities, the method further comprising

retrieving the software GUI test instances from the test queue for execution according to their associated priorities.

25. The method of Claim 17, further comprising:

5 maintaining a first queue containing higher priority software GUI test instances and a second queue containing lower priority software GUI test instances;

retrieving higher priority software GUI test instances from the first queue for execution during a first part of a testing period; and

10 retrieving lower priority software GUI test instances from the second queue for execution during a second part of the testing period.

15 26. The method of Claim 17, further comprising re-communicating instances of a software GUI test for execution against all client-server combinations, according to a rule, in response to receiving one or more test results for the software GUI test indicating failure.

20 27. The method of Claim 17, further comprising detecting when the number of software GUI test instances in the test queue is below a predefined threshold and, in response, automatically adding software GUI test instances to the test queue.

25 28. The method of Claim 17, further comprising automatically installing a current software GUI build at each distributed test execution computer at one or more appropriate times during a testing period.

29. The method of Claim 17, further comprising automatically rebooting each distributed test execution computer according to a predetermined schedule.

30 30. The method of Claim 17, further comprising automatically establishing communication between the distributed test execution computer and a test server engine when the distributed test execution computer boots up.

31. The method of Claim 17, wherein each test execution computer operates essentially as an automated test execution robot, repeatedly requesting, receiving, executing, and returning test results for software GUI test instances, automatically and without human intervention, for an extended time period.

卷之三

32. Software for conducting distributed automated software graphical user interface (GUI) testing, the software being embodied in computer-readable media and when executed operable to:

5 maintain a centralized test queue operable to store a plurality of software GUI test instances to be executed by a plurality of distributed test execution computers, each distributed test execution computer comprising a client platform and coupled to one or more server platforms, the client platforms and server platforms collectively providing a plurality of client-server combinations against which the software GUI test instances may be executed; and

10 receive, for each distributed test execution computer, a request for a software GUI test instance from a particular distributed test execution computer in response to completion of a preceding software GUI test instance by the particular distributed test execution computer;

15 retrieve, for each distributed test execution computer, a software GUI test instance from the test queue;

20 communicate, for each distributed test execution computer, the retrieved software GUI test instance to the particular distributed test execution computer for execution against a particular client-server combination using a testing component supported by the particular distributed test execution computer, the testing component operable to perform automated software GUI testing and to produce test results for such testing;

25 receive, for each distributed test execution computer, a test result for the software GUI test instance from the particular distributed test execution computer in response to execution of the software GUI test instance; and

store, for each distributed test execution computer, the received test result for reporting to one or more users.

33. The software of Claim 32, wherein at least one distributed test execution computer operates at a location geographically remote from the other distributed test execution computers and from the software.

34. The software of Claim 32, wherein the testing component is a

commercial off-the-shelf product.

35. The software of Claim 32, wherein each software GUI test instance is an instance of a software GUI test written using a test scripting language and can be executed using any of the distributed test execution computers, a software GUI test instance being executed using the particular distributed test execution computer from which the request initiating retrieval of the software GUI test instance from the test queue was received.

10 36. The software of Claim 32, further operable to:

generate a test results web page comprising test results for a plurality of software GUI test instances, including the test result for the most recently executed software GUI test instance, substantially immediately upon receiving the test result from the particular distributed test execution computer on which the most recently executed software GUI test instance was executed; and

15 communicate the test results web page for display on a user system to provide substantially real-time test results reporting.

20 37. The software of Claim 36, wherein:

each software GUI test instance is an instance of a software GUI test; and further operable to generate a test results web page comprising consolidated test results for a particular client platform, the consolidated test results indicating test results for each software GUI test for each client-server combination involving the particular client platform.

25 38. The software of Claim 36, further operable to receive a user request to execute an instance of a particular software GUI test and to insert the requested software GUI test instance into the test queue according to the user request, the user request being input by selecting the particular software GUI test using the test results web page.

30 39. The software of Claim 32, wherein at least some software GUI test

instances in the test queue have associated priorities and the software is further operable to retrieve the software GUI test instances from the test queue for execution according to their associated priorities.

5 40. The software of Claim 32, wherein the test queue comprises a first queue containing higher priority software GUI test instances and a second queue containing lower priority software GUI test instances, the software is further operable to retrieve higher priority software GUI test instances from the first queue for execution during a first part of a testing period and retrieve lower priority software
10 GUI test instances from the second queue for execution during a second part of the testing period.

15 41. The software of Claim 32, further operable to re-communicate instances of a software GUI test for execution against all client-server combinations, according to a rule, in response to receiving one or more test results for the software GUI test indicating failure.

20 42. The software of Claim 32, further operable to detect when the number of software GUI test instances in the test queue is below a predefined threshold and, in response, to automatically add software GUI test instances to the test queue.

25 43. The software of Claim 32, further comprising software associated with each distributed test execution computer and operable to automatically install a current software GUI build at each distributed test execution computer at one or more appropriate times during a testing period.

30 44. The software of Claim 32, further comprising software associated with each distributed test execution computer and operable to automatically reboot each distributed test execution computer according to a predetermined schedule.

45. The software of Claim 32, further comprising software associated with each distributed test execution computer and operable to automatically establish

communication, when the distributed test execution computer boots up, required for the distributed test execution computer to receive software GUI test instances for execution.

- 5 46. The software of Claim 32, wherein each test execution computer
operates essentially as an automated test execution robot, repeatedly requesting,
receiving, executing, and returning test results for software GUI test instances,
automatically and without human intervention, for an extended time period.

47. A system for distributed automated software GUI testing, comprising:

means for maintaining a centralized test queue operable to store a plurality of software GUI test instances to be executed by a plurality of distributed test execution computers, each distributed test execution computer comprising a client platform and coupled to one or more server platforms, the client platforms and server platforms collectively providing a plurality of client-server combinations against which the software GUI test instances may be executed; and

means for receiving a request for a software GUI test instance from each particular distributed test execution computer in response to completion of a preceding software GUI test instance by the distributed test execution computer;

means for retrieving a software GUI test instance from the test queue in response to the request from the particular distributed test execution computer;

means for communicating the retrieved software GUI test instance to the particular distributed test execution computer for execution against a particular client-server combination using a testing component supported by the particular distributed test execution computer, the testing component operable to perform automated software GUI testing and to produce test results for such testing;

means for receiving a test result for the software GUI test instance from the particular distributed test execution computer in response to execution of the software GUI test instance; and

means for storing the received test result for reporting to one or more users.

48. A system for distributed automated software graphical user interface (GUI) testing, comprising:

a centralized test queue operable to store a plurality of software GUI test instances to be executed by a plurality of distributed test execution computers, each distributed test execution computer comprising a client platform and coupled to one or more server platforms, the client platforms and server platforms collectively providing a plurality of client-server combinations against which the software GUI test instances may be executed, each software GUI test instance is an instance of a software GUI test written using a test scripting language and executable using any of the distributed test execution computers;

10 a test server engine operable to, for each distributed test execution computer:

receive a request for a software GUI test instance from the particular distributed test execution computer in response to completion of a preceding software GUI test instance by the particular distributed test execution computer;

15 retrieve a software GUI test instance from the test queue;

communicate the retrieved software GUI test instance to the particular distributed test execution computer for execution against a particular client-server combination using a testing component supported by the particular distributed test execution computer, the testing component operable to perform automated software GUI testing and to produce test results for such testing for communication to the test server engine, a software GUI test being executed using the particular distributed test execution computer from which the request initiating retrieval of the software GUI test from the test queue was received;

20 receive a test result for the software GUI test instance from the particular distributed test execution computer in response to execution of the software GUI test instance;

25 store the received test result for reporting to one or more users in a test results database; and

30 generate a test results web page comprising the test results for the plurality of software GUI test instances.

each distributed test execution computer operating essentially as an automated test execution robot, repeatedly requesting, receiving, executing, and returning results

for software GUI test instances automatically without human intervention for extended periods of time; and

a web server operable to:

access the test results database to obtain test results for a plurality of software GUI test instances; and

communicate the test results web page for display on a user system to provide substantially real-time test results reporting.

卷之三